An Efficient Load Balancing in Public Cloud using Priority based SJF Scheduling

Suchita Khare, Asst. Prof. Shatendra Dubey

Abstract— Cloud Computing enables various users to share resources and exchange information over internet. But several issues arises during the sharing of resources such as load balancing over cloud. Hence various techniques are implemented for the balancing of users during sharing of resources. Here in this paper a new and efficient technique is implemented using hybrid combination of scheduling using priority of resources access and shortest access time of the resources. The proposed technique implemented here provides efficient load balancing as well as provides less waiting time and have efficient response time.

Index Terms— CSP, FCFS, BMHA, MFTF, Priority Vector, CPU Utilization.

1 Introduction

Though Cloud Computing has a glorious future, many crucial problems still need to be solved for the realization of Inter-Cloud computing. One of these problems is load distribution with less complexity and better server utilization. It plays a very important role in the user request deployment in Inter-Cloud environment with minimum response time. It is used by Cloud service provider (CSP) in its own cloud computing platform to provide a high efficient solution for the user. Also, a inter CSP load distribution mechanism is needed to construct a low cost and infinite resource pool for the consumer. Load distribution in cloud computing is to distribute the local workload evenly to the whole cloud. In fact it has become indispensable for cloud computing. Thus Load distribution in cloud computing provides an organization with the ability to distribute application requests across any number of application deployments located in datacenters and through cloud computing providers. Since the emergence of cloud computing, with the continuous development of science and technology and advance by academia and industry, the applications of cloud computing are going on developing.

Cloud computing is moving from theory to practice [1-5]. As a novel large-scale distributed computing paradigm based on virtualized resource pool, cloud computing derived from the ever-increasing interaction and profound development of distributed computing, parallel computing, grid computing, utility computing, web services, etc. If only with accesses to cloud data centers, cloud computing could enable users all over the planet to on-demand leverage a variety of IT-related services based on a pay-per-use model, e.g. infrastructure, platform, software, etc. In addition, cloud computing also corresponds with the basic idea of green computing[6-7].Through elastic and high-scalable management of the resource pool, as well as relaxation of the terminal, cloud computing can conserve much energy and cut the cost of both user's and datacenter's down. It is one of the most promising computing paradigms in the coming low carbon economy.

The main goal of job scheduling is to achieve a high performance computing and the best system throughput. Traditional job scheduling algorithms are not able to provide scheduling in the cloud environments. According to a simple classification [8] job scheduling algorithms in cloud computing can be categorized into two main groups; Batch mode heuristic scheduling algorithms (BMHA) and online mode heuristic algorithms. In BMHA, Jobs are queued and collected into a set when they arrive in the system. The scheduling algorithm will start after a fixed period of time. The main examples of BMHA based algorithms are; First Come First Served scheduling algorithm (FCFS), Round Robin scheduling algorithm (RR), Min–Min algorithm and Max–Min algorithm. By On-line mode heuristic scheduling algorithm, Jobs are scheduled when they arrive in the system. Since the cloud environment is a heterogeneous system and the speed of each processor varies quickly, the on-line mode heuristic scheduling algorithms are more appropriate for a cloud environment. Most fit task scheduling algorithm (MFTF) is suitable example of On-line mode heuristic scheduling algorithm [8].

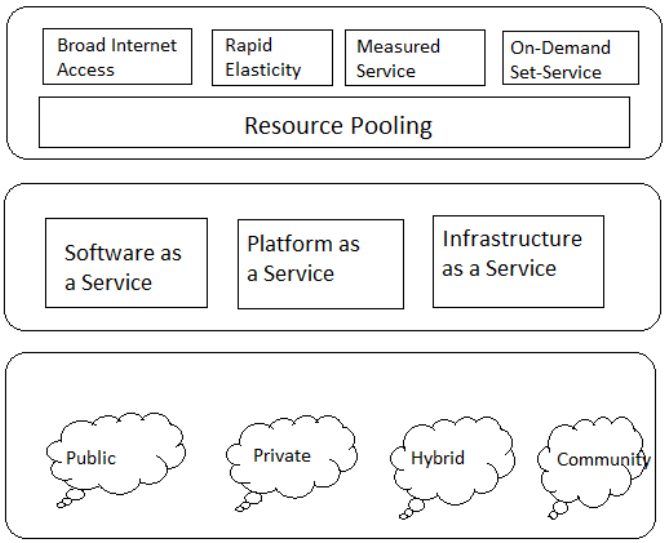

Figure 1: Basic Cloud Computing Model

2 Literature Survey

In 2013, Xu, Gaochao et al [9] presented A load balancing model based on cloud partitioning for the public cloud. The load balancing model is aimed at the public cloud which has

numerous nodes with distributed computing resources in many different geographic locations. Thus, this model divides the public cloud into several cloud partitions. When the environment is very large and complex, these divisions simplify the load balancing. The cloud has a main controller that chooses the suitable partitions for arriving jobs while the balancer for each cloud partition chooses the best load balancing strategy. The load balance solution is done by the main controller and the balancers. The main controller first assigns jobs to the suitable cloud partition and then communicates with the balancers in each partition to refresh this status information. Since the main controller deals with information for each partition, smaller data sets will lead to the higher processing rates. The balancers in each partition gather the status information from every node and then choose the right strategy to distribute the jobs.

Naimesh D. Naik and Ashilkumar R. Patel proposed Load Balancing Under Bursty Environment for Cloud Computing. Current algorithms do not consider the bursty workloads and hence it will decrease the system performance. Their proposed algorithm considers the current VM resource utilization and bursty workloads for distributing the load to each VM instances. They expect that using the proposed algorithm cloud service provider can meet the service level agreements (SLA) without purchasing additional resources. This algorithm also ensures that none of VM resources is over utilized when another one is underutilized. This will increase the system performance and provide faster response time. This will also increase the economic profit of an organization as all the resources are better utilized so there is no need for extra resources for handling the request [10].

In year 2012, Kaviani, Nima et al [11] proposed MANTICORE: A framework for partitioning software services for hybrid cloud. They discussed MANTICORE, a framework that allows software developers to analyze a monolithic (i.e single host) software service to make it suitable for hybrid cloud. In order to verify the accuracy of the cost models, they compare the execution and data transfer measurements of models generated by MANTICORE to those of real deployments for DayTrader. Since our profiling data was collected on the premise machine, the models and the real deployment for the machine on premise are identical. In a hybrid or a full cloud deployment, the models are generated by applying linear fitting techniques. They compared the generated cost models with the real deployment of the DayTrader application for settings where the deployment is fully in the cloud or is MHD [11].

In same year, P. Jamuna and R.Anand Kumar proposed Optimized Cloud Partitioning Technique to Simplify Load Balancing. This model divides the cloud environment into several partitions by making use of cloud clustering technique, helps the providers to simplify the process of load balancing. Thus this proposed technique achieves higher performance and stability in cloud. Arriving patterns of jobs are unpredictable, thus allocating and processing of many jobs over cloud environment among various node is a complex problem. And the capacity of each node also differs from each other. Hence load should be balanced among multiple nodes in order to improve the stability and system performance. The public cloud environment which has enormous nodes over large and different geographical location. Our proposed model divides the cloud environment into several partitions. This partitioning technique helps the providers to simplify the process of load balancing [12].

CloneCloud [13] optimizes for high performance and minimum resource usage for applications in mobile and embedded devices by offloading the execution to the cloud; but optimizing towards a cheaper deployment is not the focus for CloneCloud. Cloudward Bound [14] and COPE [15] optimize for cost of deployment in a hybrid setting, however there are several differences between these approaches and MANTICORE. For these approaches, partitioning and relocation of components happens at the level of application servers (or VMs) not the finer level of code entities (i.e. functions) as is the case with MANTICORE. Furthermore, none of these approaches supports context sensitivity. Finally, while COPE does not account for latency, Cloudward Bound enforces the accepted latency by defining an upper limit constraint, whereas MANTICORE allows for software developers to decide about their preferred cost-to-latency ratios.

3 PROPOSED METHODOLOGY

The proposed methodology implemented here for the scheduling of the resources in cloud computing using hybrid combinatorial method of priority based Shortest Job First.

FIRST COME FIRST SERVER

1. First of all create Cloud Simulation environment.
2. set the number of user 'Ui' and a Number of resources 'Ri' to access.
3. Schedule allocation of resources 'Ri' to Users 'Ui' using First Come First Server algorithm.
4. The Users that request the Resource First is allocated the resource first.
5. It is also called FIFO.

SHORTEST JOB FIRST

1. First of all create Cloud Simulation environment.
2. set the number of user 'Ui' and a Number of resources 'Ri' to access.
3. Schedule allocation of resources 'Ri' to Users 'Ui' using Shortest Job First algorithm.
4. Scheduling of Resources can be done on the basis of their shortest duration time.

PRIORITY BASED SCHEDULING

1. First of all create Cloud Simulation environment.
2. set the number of user 'Ui' and a Number of resources 'Ri' to access.

3. Schedule allocation of resources 'Ri' to Users 'Ui' using priority based scheduling algorithm.
4. Scheduling of Resources can be done on the basis of priority of resources.

PROPOSED SCHEDULING

The proposed methodology implemented here consists of combining priority based scheduling and Shortest Job First Scheduling.
1. First of all create Cloud Simulation environment.
2. set the number of user 'Ui' and a Number of resources 'Ri' to access.
3. Schedule allocation of resources 'Ri' to Users 'Ui' using priority but checks whether the r
Source takes shortest duration time to execute or not.

ALGORITHM

The proposed methodology uses the combination of two scheduling techniques Shortest Job First and Priority based Scheduling. The processing can be applied for the public as well as Hybrid Cloud. The formal algorithms steps of the proposed methodology are given below:

Notation	Description
Ui	Various Users of the cloud
DCi	Data Centers
UBi	Broker of the cloud
Bi	Burst Time for the Job Ji
Ji	Sequence of Jobs
Ti	Process time
Pi	Priority of Job Ji

Table 1 Various Notations used in Algorithm

1. If 'N' is the number of requests to send from 'Ui; users of the cloud 'C' with 'Bi' burst time of each of the user 'Ui' to the data Centers 'DCi' through brokers 'UBi'.
2. If 'T1, T2, ……Tn' is the various burst time from various users "Ui' for the request of the Jobs 'Ji'.
3. If 'P1,P2,……Pn' be the priority of various jobs 'Ji' for the request.
4. Compute Priority vector for all d matrices using $A_w = \gamma_{maxw}$
5. Make a matrix with priority vector using

$$\Delta = [w^1 w^2 \ldots . w^d]$$

6. Compute 'C' for the consistent comparison matrix.
7. Compute PVS which is a vector included value of priority of jobs.
8. Check the 'Ji' having highest priority from 'Pi'.
9. Now also check the burst time 'T' of the job 'Ji' having highest priority 'Pi'.
10. If burst time 'Ti' is very less then execute the scheduling of the job 'Ji'.
11. Otherwise the job having shortest burst time 'Ti' is executed.

FLOW CHART

The figure shows flow chart of the proposed methodology. It consists of number of inputs from the users of the cloud along with their burst time. Now the jobs are send to the broker for scheduling. Now at the broker side burst time of each of the user is checked and their priority, if the priority of the user is more than that job is executed first but if the highest priority job contains more burst time than this job can't be executed first.

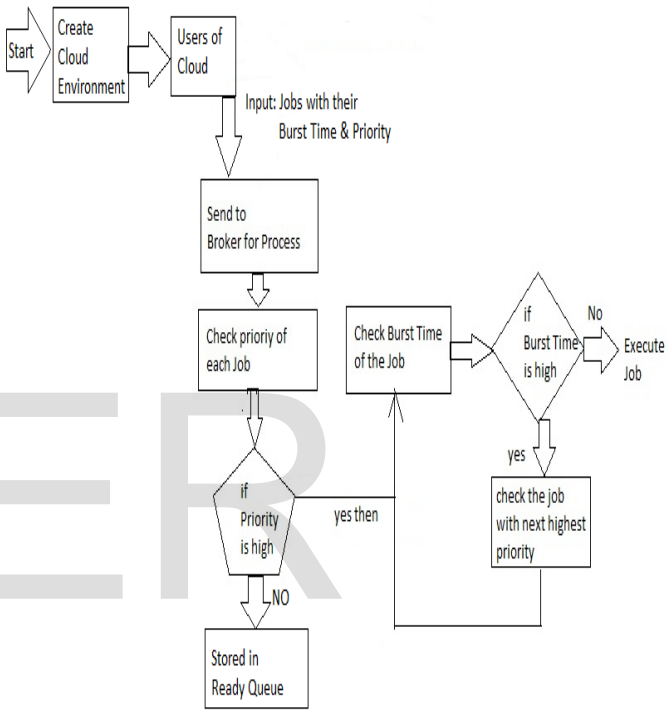

Figure 2 Flow Chart of the proposed methodology

Since Cloud environment contains a number of cloudless also known as users and brokers can be classified as local brokers and global brokers and number of datacenters processed by the virtual machines. The users of the cloud can send the data to the data centers through brokers. In the cloud environment users can also send the request or job to process, hence scheduling is done between users so that the resources can be utilized easily and quickly. The figure shown below is the flow chart of the proposed methodology. The various request of the user is first send to the local brokers where broker checks thee priority of each job and the job having highest priority will be processed first but before that processing time is checked if is more than the other job than the job having less time and high priority is processed first and likewise other jobs are processed.
When scheduling is done and all the jobs are processed successfully, then a gantt chart is created for the waiting time and turn around time and response time of each of the users job request so that the chances of deadlock is less and CPU utilization increases.

4 Result Analysis

The table shown below is the comaprison of existing framework and the proposed framework for the energy consumption .

	Energy Consumption	
Data Size (L bits)	**Existing**	**Proposed**
0	0	0
500	100	200
1000	800	500
1500	7300	2000
2000	10000	6500
2500	12000	8294

Table 2 Comparison of Energy Consumption

The table shown below is the analysis of the total bits to be transmitted over a certain delay time of the communication inn mobile devices.

Delay Deadline	**Existing Work**	**Proposed Work**
0	0	240
50	200	439
100	400	591
150	600	783
200	800	940
250	1000	1158

Table 3 Comparison of Delay Deadline

The figure shown below is the comaprison of existing framework and the proposed framework for the energy consumption.

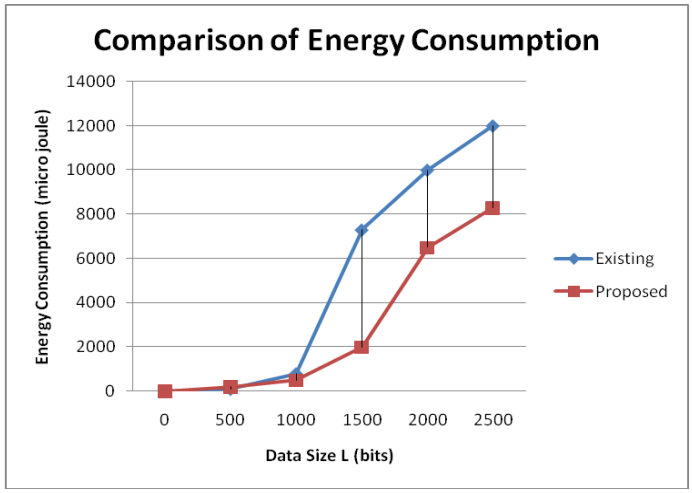

Figure 3 Comparison of Energy Consumption

The figure shown below is the analysis of the total bits to be transmitted over a certain delay time of the communication inn mobile devices.

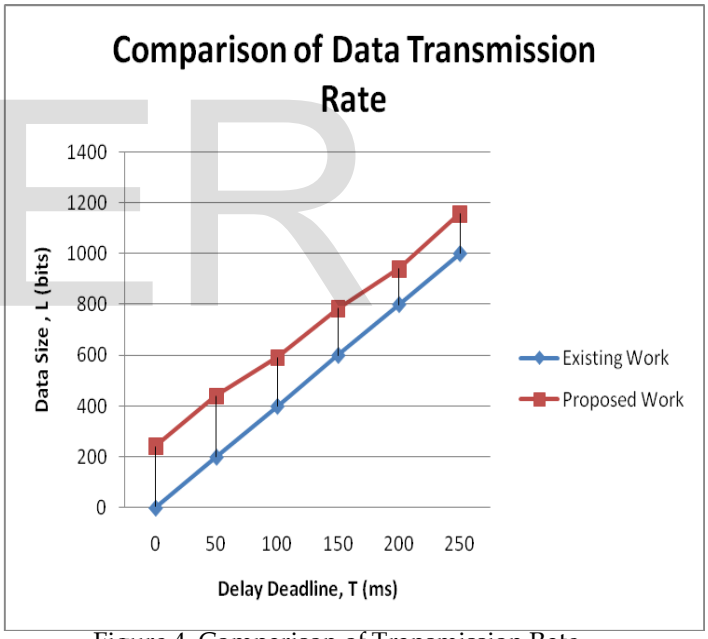

Figure 4. Comparison of Transmission Rate

5 Conclusion

The framework implemented here for the energy optimal energy consumption for the mobile cloud computing in wireless channel provides less energy consumption. Also the technique implemented here provides more data bits transmitted in mobile devices with less storage and energy consumption. The methodology implemented here reduces the communication overhead for the transmitted bits, hence latency reduces.

The result analysis shows the performance of the proposed methodology. A Simulation is done on various scheduling technique, but the proposed methodology provides efficient waiting time and turnaround time.

The proposed methodology when implemented on the cloud simulation environment provides only about 300.06 overall average response time in the cloud simulation environment.

While in terms of computational cost will only provide about 0.57$.

When comparison is done between First Come First Server Scheduling and Shortest Job First Scheduling and proposed scheduling, the proposed methodology provides less processing time among all.

REFERENCES

[1] A model of Virtual Resource Scheduling in Cloud Computing and Its Solution using EDAs", JDCTA: International Journal of Digital Content Technology and its Applications, Vol. 6, No. 4, pp. 102-113, 2012.

[2] [2] LIANG Quan, HE Wen-wu, LONG Jian-hui, "A Real-time Service Model Based on Cloud Computing in the Next Generation Internet", IJACT: International Journal of Advancements in Computing Technology, Vol. 4, No. 9, pp. 280-287, 2012.

[3] [3] Guo Xinzhi, Niu Liping, "Research on Risk Assessment Model of Information System Based on Cloud Environment", JCIT: Journal of Convergence Information Technology, Vol. 7, No. 11, pp. 288-296, 2012.

[4] [4] Rui Wang, Qinghua Bai, Jia You, Hualing Liu, "Research on Semantic Metadata Model for Information Resource Management under Cloud Computing Environment", JCIT: Journal of Convergence Information Technology, Vol. 7, No. 21, pp. 18-26, 2012.

[5] [5] Kai-jian Liang, Quan Liang, "A Dynamic Method of Model Constructing For Cloud Computing", AISS: Advances in Information Sciences and Service Sciences, Vol. 4, No. 17, pp. 174-182, 2012.

[6] [6] Wang Juan Li Fei Chen Aidong, "An Improved PSO based Task Scheduling Algorithm for Cloud Storage System", AISS: Advances in Information Sciences and Service Sciences, Vol. 4, No. 18, pp. 465-471, 2012.

[7] [7] Dung-Hai Liang, Peirchyi Lii, Dong-Shong Liang, "Risk Management of Land Use on Cloud Computing", JCIT: Journal of Convergence Information Technology, Vol. 7, No. 1, pp. 122-129, 2012.

[8] [8] A new Class of Priority-based Weighted Fair Scheduling Algorithm, Physics Procedia ,33 (2012) 942 – 948.

[9] [9] Xu, Gaochao, Junjie Pang, and Xiaodong Fu. "A load balancing model based on cloud partitioning for the public cloud." IEEE Tsinghua Science and Technology, Vol. 18, no. 1, pp. 34-39, 2013.

[10] [10] Naimesh D. Naik and Ashilkumar R. Patel "Load Balancing Under Bursty Environment for Cloud Computing", International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181, Vol. 2, Issue 6, pp. 17 – 26, June – 2013.

[11] [11] Kaviani, Nima, Eric Wohlstadter, and Rodger Lea. "MANTICORE: A framework for partitioning software services for hybrid cloud." In *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*, pp. 333-340, 2012.

[12] [12] P. Jamuna and R.Anand Kumar " Optimized Cloud Partitioning Technique to Simplify Load Balancing", International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X, Volume 3, Issue 11, pp. 820 – 822, November 2013.

[13] [13] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: elastic execution between mobile device and cloud." in Proc. of the European Symposium on Operating Sytems (EuroSys), 2011.

[14] [14] M. Hajjat, X. Sun, Y.-W. E. Sung, D. Maltz, S. Rao, K. Sripanidkulchai, and M. Tawarmalani, "Cloudward bound: planning for beneficial migration of enterprise applications to the cloud," in SIGCOMM, 2010.

[15] [15] C. Liu, B. T. Loo, and Y. Mao, "Declarative automated cloud resource orchestration," in Proc. of the Symposium on Cloud Computing, 2011.